

# Programação 1 e Inteligência Artificial – Trabalho Prático 2/3

Especialização em Desenvolvimento de Jogos Digitais - PUCRS

Professores: Sílvia Moraes, Marcelo Cohen e Bruno Ferreira

## Objetivo do Trabalho

Desenvolver o *remake* de um dos seguintes jogos clássicos:

### **Moon Patrol (Arcade)**

[https://en.wikipedia.org/wiki/Moon\\_Patrol](https://en.wikipedia.org/wiki/Moon_Patrol)

### **Commando (Arcade)**

[https://en.wikipedia.org/wiki/Commando\\_\(video\\_game\)](https://en.wikipedia.org/wiki/Commando_(video_game))

### **Double Dragon (Arcade)**

[https://en.wikipedia.org/wiki/Double\\_Dragon](https://en.wikipedia.org/wiki/Double_Dragon)

### **Puzzle Bobble (Arcade)**

[https://en.wikipedia.org/wiki/Puzzle\\_Bobble](https://en.wikipedia.org/wiki/Puzzle_Bobble)

### **Boulder Dash (NES)**

[https://en.wikipedia.org/wiki/Boulder\\_Dash](https://en.wikipedia.org/wiki/Boulder_Dash)

### **Road Fighter (NES)**

[https://en.wikipedia.org/wiki/Road\\_Fighter](https://en.wikipedia.org/wiki/Road_Fighter)

### **Gradius (NES)**

[https://en.wikipedia.org/wiki/Gradius\\_\(video\\_game\)](https://en.wikipedia.org/wiki/Gradius_(video_game))

### **Metroid (NES)**

[https://en.wikipedia.org/wiki/Metroid\\_\(video\\_game\)](https://en.wikipedia.org/wiki/Metroid_(video_game))

### **Yie Ar Kung-Fu (MSX)**

[https://en.wikipedia.org/wiki/Yie\\_Ar\\_Kung-Fu](https://en.wikipedia.org/wiki/Yie_Ar_Kung-Fu)

Antes de escolher um dos jogos, veja vídeos e **jogue-os** em emuladores. Dê preferência para as versões das plataformas conforme especificadas na lista.

## Desenvolvimento

Este trabalho poderá ser desenvolvido individualmente ou em duplas. A implementação deve ser feita em Unity com C# e rodar na plataforma Windows 10. Desenvolvimento do trabalho será feito durante as próximas aulas, com o apoio dos professores e troca de experiências com os colegas. Porém, é essencial que os alunos dediquem tempo extraclasse para a sua conclusão.

**O trabalho deverá obrigatoriamente ser desenvolvido com a utilização do git para controle de versão e hospedado em um repositório público (gratuito) do GitHub ou similar.** Caso nunca tenha utilizado git ou controle de versão, ou possua dificuldades, peça auxílio para os colegas ou professores durante a aula ou consulte o livro oficial gratuitamente na internet:

<https://git-scm.com/book/pt-br/v1/>

O trabalho é subdividido em duas etapas distintas, cada qual valendo 50% da nota do trabalho, conforme a seguir:

### **Etapas 1 – Protótipos: (Trabalho 2)**

1. Definir quais são as três mecânicas fundamentais do jogo escolhido;
2. Prototipar estas três mecânicas fundamentais, de preferência em cenas separadas na Unity. O objetivo dos protótipos é testar cada uma das mecânicas isoladamente do resto do jogo;
3. Etapas 1 a ser apresentada em aula no dia 08/01/2021.

### **Etapas 2 – Versão Completa: (Trabalho 3)**

1. Implementar pelo menos um nível completo (ou equivalente) do jogo escolhido;
2. O remake deve ter a jogabilidade mais próxima o possível do jogo original. Faça uma “engenharia reversa”, analisando o tamanho dos elementos com relação à tela, cronometre para descobrir a velocidade dos personagens e estude minuciosamente o comportamento dos elementos do jogo;
3. Os gráficos e sons não precisam ser os mesmos do jogo original (podem ser utilizados quaisquer substitutos). Porém, é bastante comum encontrar os assets dos originais na Internet. Evite perder tempo fazendo arte neste trabalho, pois as disciplinas são de programação;
4. Etapas 2 a ser apresentada em aula no dia 29/01/2021.

## Avaliação

A avaliação considerará:

1. Participação de cada aluno na elaboração do projeto. Isto pode resultar em notas diferentes para os colegas no mesmo grupo. Por isso é fundamental que o controle de versão esteja configurado corretamente para que possa ser avaliada a contribuição de cada aluno;
2. A avaliação será **penalizada** caso não seja utilizado controle de versão ou este não esteja configurado corretamente com usuários separados para cada aluno;
3. Entrega das etapas do trabalho, com o máximo dos requisitos cumpridos e dentro dos prazos estipulados;
4. Qualidade da implementação (código, arquivos de dados e organização do projeto) e criatividade para encontrar soluções;
5. Similaridade com o jogo original, na fase escolhida;
6. Apresentação em aula das duas etapas. Caso algum aluno não possa comparecer nas datas pré-estabelecidas, outra data deve ser negociada com os professores;
7. Não é permitido utilizar nenhuma biblioteca ou plugin externo para programação da lógica de gameplay ou desenho de imagens. Todo o código deve ser desenvolvido pelo aluno. É permitido utilizar Assets gráficos e sonoros gratuitos (por exemplo, disponíveis na Unity Asset Store).

***Bom trabalho e divirtam-se!***